wizehive

Overcoming Software Overload

If you often find your foundation conforming to the limitations of its software and conducting its processes in inefficient ways, then it may be time to take a step back and reassess your software strategy.

In this whitepaper, we will be reviewing the results of a survey on the most popular software solutions used across many foundations. We will then explore tools and strategies your foundation can use to better integrate and customize that software, thus maximizing its benefits.

The Survey

In our survey, we asked our foundation customers about the amount of software they used, what that software was, and what their pain points were. As we received the responses, we noticed a lot of commonality between the problems reported and the solutions available.

Our average customer reports using 6 software packages on a regular basis.

That's a lot of jumping around between software, but we believe the actual number is even higher! The average organization is so inundated with software — including specialized software that is only used by single individuals or departments — that it has become practically impossible to remember and catalog every tool that is depended upon.

The most popular software packages that our customers report are centered around very common needs — email, finance, document collaboration and synchronization, project management, application/form processing, constituent relationship management, reporting, signatures, etc. It is common to find a separate software package being used for each need.

So we asked a question:

"If you could wave a magic wand to make one change that would make you more productive at work, what would you change?"

The responses overwhelmingly spoke to automation and consolidation of processes.

As we see it, the biggest problem isn't the number of software packages, but rather a lack of future-proofing when selecting and implementing new software. As organizations put new software in place, they often do so without streamlining their processes and automating the connections between the new software and the old software. This results in frustrated teams that are less likely to want to fully utilize the potential of the software.

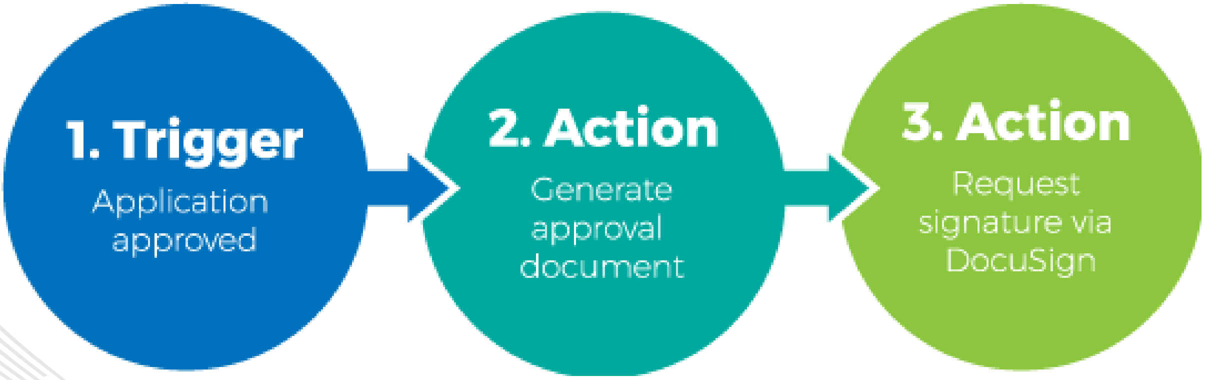
In addition, new solutions are often deployed without consideration for future needs. For example, organizations may choose not to make strategic investments in constituent relationship management software due to the integration limitations of legacy financial software. When one piece of the software chain is inflexible, the lack of flexibility spreads like a virus throughout the entire chain, leading to inefficient processes.

The solution lies in looking for and leveraging integration and customization capabilities.

Integrations

Every integration you create between two or more software packages can do one or more of: save administrator time, save constituent time, reduce manual data entry and errors, and even maintain legal compliance.

An integration typically starts with a trigger which then sets off one or more actions. For example, let's say your process requires that you obtain a grantee's electronic signature acknowledging the approval of a grant application. If you're doing this manually, your program administrator likely types up an approval document and then mails it to the grantee for signature. But with an automated integration, the administrator marks the application as approved in a grants management system, and that event is set as a trigger to then generate an approval document automatically and upload it to a service like DocuSign to request a signature from the applicant. The idea is that no one piece of software is handling everything. Each software package specializes in what it does best, yet no time is wasted and no accidents are made manually handling processes at the connection points.



The only reason this example works, though, is because the grants management system and the electronic signature software both have integration capabilities. This is our first lesson in future-proofing: when choosing software, look for integration capabilities that will make the software stickier and more likely to remain relevant as your foundation's processes evolve.

This is our first lesson in future-proofing: When choosing software, look for integration capabilities that will make the software stickier and more likely to remain relevant as your foundation's processes evolve.

Specialized Integration Tools

Going a step further, many modern software packages work with centralized integration tools. If a software package is like a phone number, then these centralized integration tools are switchboard operators. One such tool is Zapier, which allows organizations to connect over 750 web-based services together from one simple and convenient interface.

This is our second lesson in future-proofing: Support for a handful of important integrations is great, but cooperation with integration standards is much better.

Even the best software falls victim to the 80/20 rule — solving 80% of an organization's problems while leaving 20% left to be desired.

Only the latter can ensure the continued relevance of a tool as the software ecosystem around it evolves.

Sometimes, though, integrations just aren't possible or aren't enough. While Zapier supports 750 services, 750 is still a subset of all software. And while integrations help ease the burden of moving data efficiently between systems, they don't solve other common concerns. For example, we often find that even the best software falls victim to the 80/20 rule - solving 80% of an organization's problems while leaving 20% left to be desired. Wouldn't it be nice if there were a way to supplement the 20%?

Customizability

Customizable software is the second most important aspect of future-proofing.

Note that customizability is different than configurability. A majority of systems offer configurability - for example, a settings panel that allows you to control the size of fonts, add signatures, etc, by pointing and clicking. But with customization options, custom code can be written to extend the platform in ways that are not accounted for out of the box. This code can typically be written in a matter of hours, and the work can be completed by technical folks on your own staff, or by your software vendor's partners or employees. Thus, customizability is a complement to configuration.

Imagine using a piece of software for five years and then realizing it needs a new piece of functionality to match a new process. You could ask the software company to add it, but that may not be a priority for them.

This is our third lesson in future-proofing: When you select a platform that allows you to customize it, then you don't need to worry much about how your needs will evolve. The ability to customize is an insurance policy against future needs.

A Note on APIs

When selecting for customization, it is important to look for systems that have rich APIs — specifically, at least one API that allows for data to flow into and out of the system, and another API that allows the user interface to be enhanced with new screens and widgets. These APIs act as evidence that a software vendor takes customization capabilities seriously.

Moving forward

So remember, when selecting your next software package, make it a point to inquire about integrations and customizations. And keep in mind that your existing systems can likely integrate with each other and be customized more than you may know. Be sure to leverage these capabilities so that you are maximizing your software's benefits and setting your foundation up for success.

To learn more about Zengine's ability to improve the experience for your members, please visit wizehive.com.